## **PathAddExtension**

Return value buffer must be large enough to store returned path

Sean Barnum, Cigital, Inc. [vita<sup>1</sup>]

Copyright © 2007 Cigital, Inc.

2007-04-02

## Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5178 bytes

A44 1 G 4	D.d. C	C : 11		
Attack Category	Path spoofing or confusion problem			
	Malicious Input			
<b>Vulnerability Category</b>	Buffer Overflow			
	Unconditional	Unconditional		
<b>Software Context</b>	Filename Manageme	Filename Management		
Location	• shlwapi.h	• shlwapi.h		
Description	The in/out buffer of the function PathAddExtension() must be large enough to hold the returned value.			
	The PathAddExtension() routine adds an arbitrarily specified file extension to a file if and only if the file doesn't already have an extension. Does nothing to a NULL string. Will probably crash on a NULL pointer.			
	The first parameter, pszPath, must be at least MAX_PATH characters in length to ensure that it is large enough to hold the returned string.			
APIs	<b>Function Name</b>	Comments		
	PathAddExtension	stores args 0+1 >> argo		
	PathAddExtensionA	ASCII		
	PathAddExtensionW	Unicode		
	ATLPath::AddExtension	Overloaded wrapper of PathExtension		
Method of Attack	variable is not long enough. Since the extension can be most dangerous when the user input. In other situation the attacker to fill the sour allow the 3+ character fixed.	Attacker can cause a buffer overflow if the path variable is not long enough to hold the variable. Since the extension can be of arbitrary length, this is most dangerous when the extension is provided by user input. In other situations, it may be possible for the attacker to fill the source buffer to a limit, then allow the 3+ character fixed entension to over run the buffer, possibly connecting it to other data.		
<b>Exception Criteria</b>				

<sup>1.</sup> http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

PathAddExtension 1

Solutions	Solution Applicability	Solution Description	Solution Efficacy	
		Ensure that destination buffer is at least MAX_PATH characters in length.	Effective.	
Signature Details	LPTSTR pszPath	BOOL PathAddExtension( LPTSTR pszPath, LPCTSTR pszExtension );		
Examples of Incorrect Code	TCHAR buffe TEXT("file" small LPSTR lpStr lpStr1 = bu TCHAR F_Ext LPCTSTR lpS lpStr3 = F_ int ret_1 =	<pre>TCHAR buffer_1[9] = TEXT("file"); // Buffer may be too small LPSTR lpStr1; lpStr1 = buffer_1; TCHAR F_Ext[] = TEXT(argv[1]); LPCTSTR lpStr3; lpStr3 = F_Ext; int ret_1 = PathAddExtension(lpStr1, lpStr3);</pre>		
Examples of Corrected Code	TCHAR buffe TEXT("file" LPSTR lpStr lpStr1 = bu TCHAR F_Ext LPCTSTR lpS lpStr3 = F_ int ret_1 =	<pre>TCHAR buffer_1[MAX_PATH] = TEXT("file"); LPSTR lpStr1; lpStr1 = buffer_1; TCHAR F_Ext[] = TEXT(argv[1]); LPCTSTR lpStr3; lpStr3 = F_Ext; int ret_1 = PathAddExtension(lpStr1, lpStr3);</pre>		
Source Reference	url=/library/	http://msdn.microsoft.com/library/default.asp? url=/library/en-us/shellcc/platform/shell/ reference/shlwapi/path/pathaddextension.asp²		
Recommended Resource				
Discriminant Set	<b>Operating Syste</b>	em • W	indows	
	Languages	• C	++	

## Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about "Fair Use," contact Cigital at copyright@cigital.com<sup>1</sup>.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

PathAddExtension 3

<sup>1.</sup> mailto:copyright@cigital.com